BIAX Corporation v. Intel
Civil Action No. 2:05-cv-184-TJW

EXHIBIT 3
(PART 1)

US005517628A

United States Patent [19]

Morrison et al.

[11] **Patent Number: 5,517,628**

[45] **Date of Patent: May 14, 1996**

[54] **COMPUTER WITH INSTRUCTIONS THAT USE AN ADDRESS FIELD TO SELECT AMONG MULTIPLE CONDITION CODE REGISTERS**

[75] Inventors: **Gordon E. Morrison**, Denver; **Christopher B. Brooks**; **Frederick G. Gluck**, both of Boulder, all of Colo.

[73] Assignee: **Biax Corporation**, Palm Beach Gardens, Fla.

[21] Appl. No.: **254,687**

[22] Filed: **Jun. 6, 1994**

Related U.S. Application Data

[63] Continuation of Ser. No. 93,794, Jul. 19, 1993, abandoned, which is a continuation of Ser. No. 913,736, Jul. 14, 1992, abandoned, which is a continuation of Ser. No. 560,093, Jul. 30, 1990, abandoned, which is a division of Ser. No. 372,247, Jun. 26, 1989, Pat. No. 5,021,945, which is a division of Ser. No. 794,221, Oct. 31, 1985, Pat. No. 4,847,755.

[51] **Int. Cl.**[6] **G06F 9/06**

[52] **U.S. Cl.** **395/375**

[58] **Field of Search** 395/375

[56] **References Cited**

U.S. PATENT DOCUMENTS

3,611,306	10/1971	Reigel .	
3,771,141	11/1973	Culler .	
4,104,720	8/1978	Gruner .	
4,109,311	8/1978	Blum et al. .	
4,153,932	5/1979	Dennis et al. .	
4,181,936	1/1980	Kober .	
4,200,912	4/1980	Harrington et al.	364/200
4,228,495	10/1980	Bernhard .	
4,229,790	10/1980	Gilliland et al.	364/DIG. 1
4,241,398	12/1980	Caril .	

(List continued on next page.)

OTHER PUBLICATIONS

Bernhard, "Computing at the Speed Limit", IEEE Spectrum, Jul. 1982, pp. 26–31.

Davis, "Computer Architecture", IEEE Spectrum, Nov. 1983, pp. 94–99.

Dennis, "Data Flow Supercomputers", Computer, Nov. 1980, pp. 48–56.

Fisher et al., "Measuring the Parallelism Available for Very Long Instruction, Word Architectures", IEEE Transactions on Computers, vol. C–33, No. 11, Nov. 1984, pp. 968–978.

Fisher et al., "Microcode Compaction: Looking Backward and Looking Forward", National Computer Conference, 1981, pp. 95–102.

Fisher, A. T., "The VLIW Machine: A Multiprocessor for Compiling Scientific Code ", Computer, 1984, pp. 45–52.

(List continued on next page.)

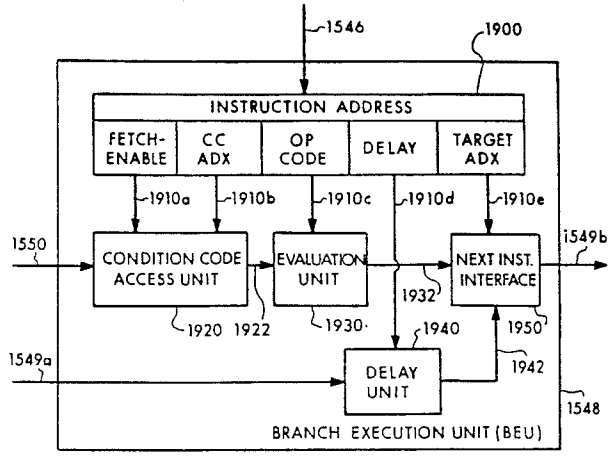
Primary Examiner—Thomas M. Heckler

Attorney, Agent, or Firm—Fish & Richardson

[57] **ABSTRACT**

The invention features a computer with a condition code register file (the condition code register file is distinct from the computer's general purpose register file). The condition code register file has a plurality of addressable condition code registers. The computer executes condition-setting instructions that each produce a condition code value for storage in one of the condition code registers, and conditional branch instructions that branch to a target based on analysis of a condition code value from one of the condition code registers. The condition code registers are directly addressable by condition code address fields of the instructions. The invention finds primary expression in one of two embodiments (or in both simultaneously): either (a) at least some of the condition-setting instructions contain a direct address field that selects one, from among the plurality of the condition code registers into which the condition code value is to be stored, or (b) at least some of the conditional branch instructions contain a direct address field that selects one, from among the plurality of the condition code registers from which a condition code value is to be selected for analysis.

30 Claims, 17 Drawing Sheets



U.S. PATENT DOCUMENTS

4,247,894	1/1981	Beismann et al.	364/200
4,250,546	2/1981	Boney et al.	364/DIG. 1
4,270,167	5/1981	Koehler et al. .	
4,334,268	6/1982	Boney et al.	364/DIG. 1
4,338,661	7/1982	Tredennick et al.	364/DIG. 1
4,342,078	7/1982	Tredennick et al.	364/200
4,430,707	2/1984	Kim .	
4,435,758	3/1984	Lorie et al. .	
4,466,061	8/1984	DeSantis .	
4,468,736	8/1984	DeSantis .	
4,514,807	4/1985	Nogi .	
4,532,589	7/1985	Shintani et al.	364/200
4,574,348	3/1986	Scallon .	
4,598,400	7/1986	Hillis	370/60
4,833,599	5/1989	Colwell et al.	364/200

OTHER PUBLICATIONS

Fisher et al., "Using an Oracle to Measure Potential Parallelism in Single Instruction Stream Programs", IEEE No. 0194–1895/81/0000/0171, 14th Annual Microprogramming Workshop, Sigmicro, Oct., 1981, pp. 171–182.

Requa et al.; "The Piecewise Data Flow Architecture: Architectural Concepts"; IEEE Transactions on Computers; vol. C–32 No. 5, May 1983, pp. 425–438.

J. R. Vanaken et al., "The Expression Processor," IEEE Transactions on Computers, C–30, No. 8, Aug., 1981, pp. 525–536.

Chang et al.; "801 Storage Architecture and Programming"; ACM Transactions on Computer Systems; 6:28–50; 1988.

Colwell et al.; "A VLIW Architecture for a Trace Scheduling Compiler"; ACM; 1987.

Ellis, John R.; "Bulldog: A Compiler for VLIW Architectures"; MIT Press; 1986; Originally Published as a Yale University Doctoral Dissertation; 1985.

Gross et al.; "Optimizing Delayed Branches"; IEEE; 114–120; 1982.

Hagiwara, et al.; "A Dynamically Microprogrammable Computer With Low–Level Parallelism"; IEEE Transactions on Computers; C–29:577–594; 1980.

Heinrich et al.; "Including the R4400 MIPS R4000 Microprocessor R4000 User's Manual"; MIPS Technologies Inc.; 1993.

Hennessy et al., "The MIPS Machine"; Proceedings of IEEE Compcon; 2–7; 1982.

Hennessy et al.; "Postpass Code Optimization of Pipeline Constraints"; ACM Transactions on Programming Languages and Systems; 5:422–448; 1983.

Hennessy; "VLSI Processor Architecture"; IEEE; c–33:1221–1246; 1984.

Hennessy; "VLSI RISC Processors"; VLSI Systems Design; 22–32; 1985.

IBM; "PowerPC™ 601, RISC Microprocessor User's Manual"; IBM and Motorola; 1991 and 1993.

Intel Corporation; "MCS–80 User's Manual (With Introduction to MCS–85™)"; Oct. 1977.

McDowell Charles E.; "A Simple Architecture for Low–Level Parallelism"; Proceedings of 1983 International Conference on Parallel Processing; 472–477; 1983.

McDowell Charles E.; "SIMAC:A Multiple ALU Computer"; Dissertation Thesis; Univ. of California; San Diego; (111 pages); 1983.

McDowell et al.; "Processor Scheduling for Linearly Connected Parallel Processors"; IEEE Transactions on Computers; c–35:632–639; Jul. 1986.

Motorola; "MC68030 Enhanced 32–Bit Microprocessor User's Manual Second Edition"; 1989.

Patterson et al.; "The Case for the Reduced Instruction Set Computer"; Computer Architecture News; 8:132–191; 1980.

Patterson David A.; "Microprogramming"; Scientific American; 248:244; 1983.

Patterson David A.; "Reduced Instruction Set Computers"; Communications of the ACM; 28:8–21; 1985.

Radin George; "The 801 Minicomputer"; Proceedings of ACM Symposium on Architectural Support for Programming Languages and Operating Systems; 10:39–47, Mar. 1982.

Sites et al.; "Alpha Architecture Reference Manual"; Digital Press; 1992.

Tomita et al.; "A User–Microprogrammable Local Host Computer With Low–Level Parallelism"; ACM 0149–7111/83/0600/0151; 151–159; 1983.

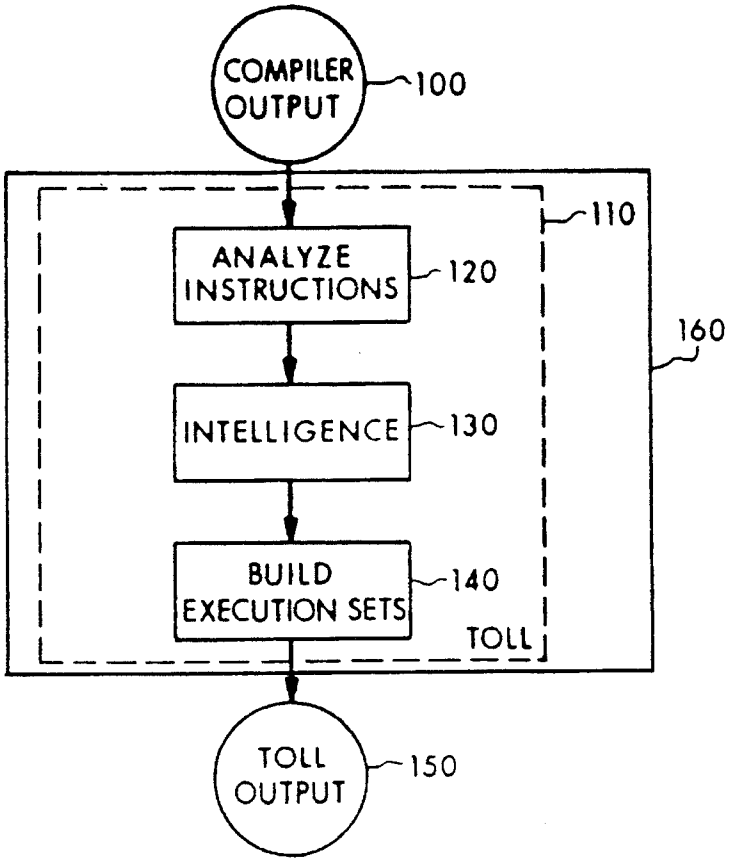


FIG. 1

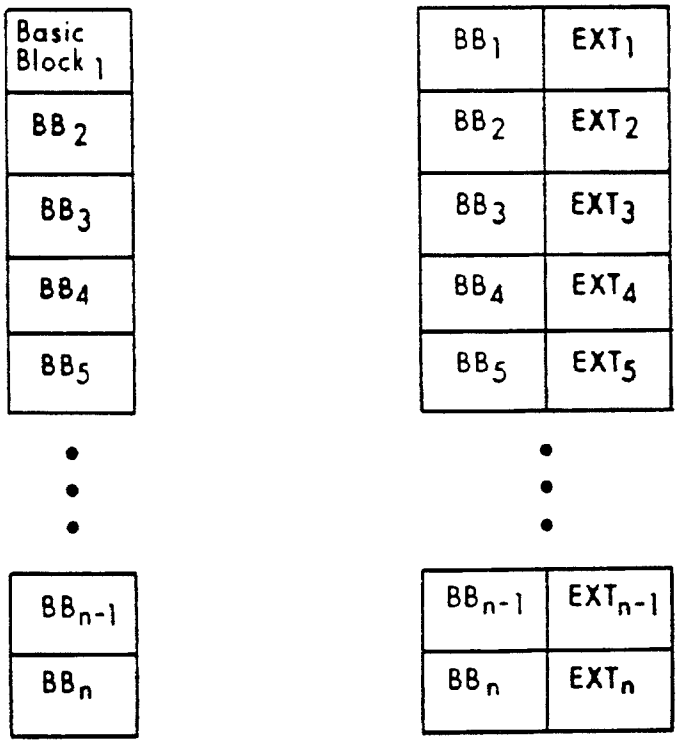


FIG. 2
PRIOR ART

FIG. 3.

FIG. 4

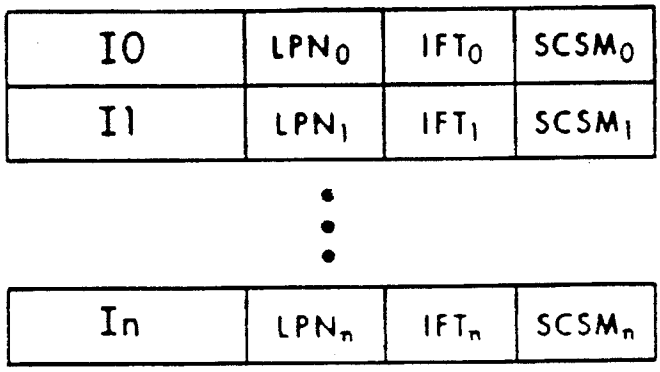
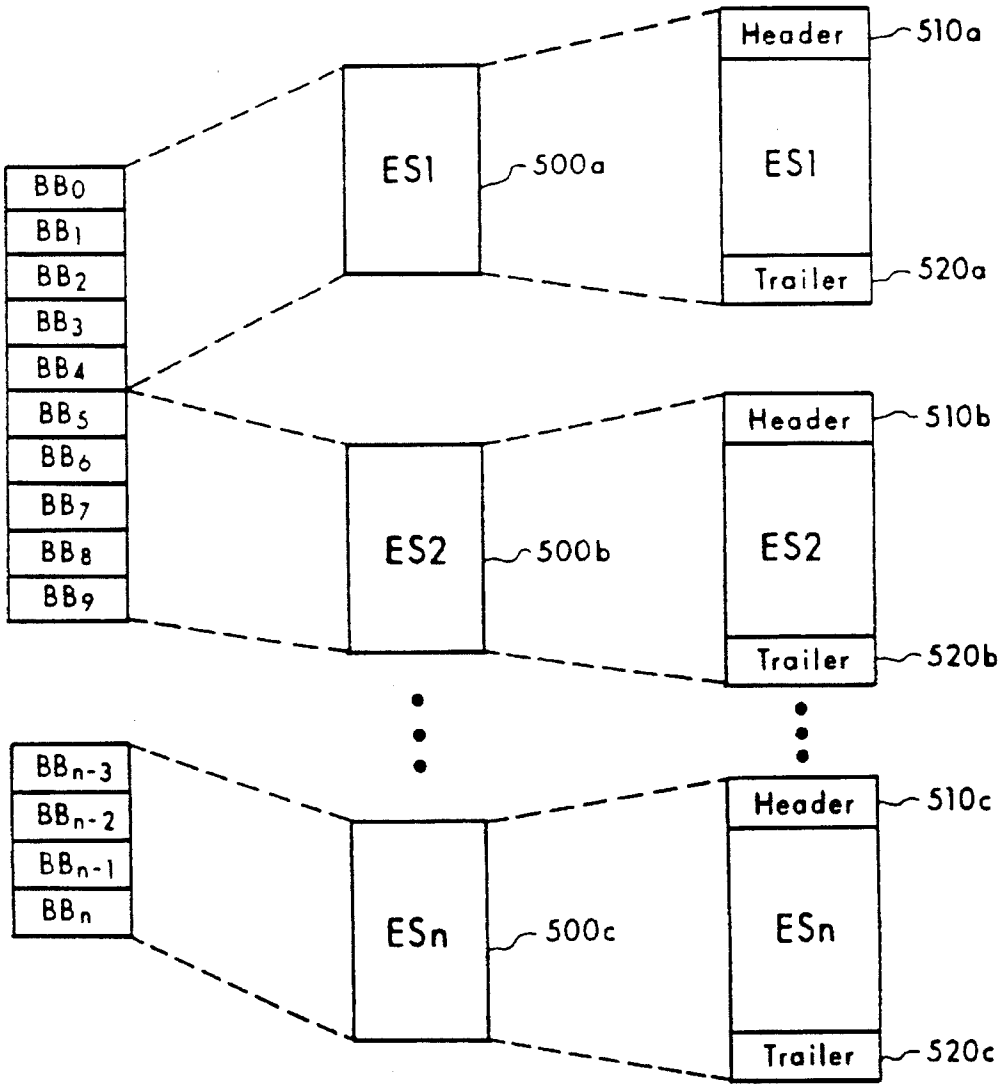

IO
LPN0
IFT0
SCSM0
I1
LPN1
IFT1
SCSM1
In
LPNn
IFTn
SCSMn

FIG. 5


BB0
BB1
BB2
BB3
BB4
BB5
BB6
BB7
BB8
BB9
BBn-3
BBn-2
BBn-1
BBn
ES1
500a
ES2
500b
ESn
500c
Header
510a
ES1
Trailer
520a
Header
510b
ES2
Trailer
520b
Header
510c
ESn
Trailer
520c

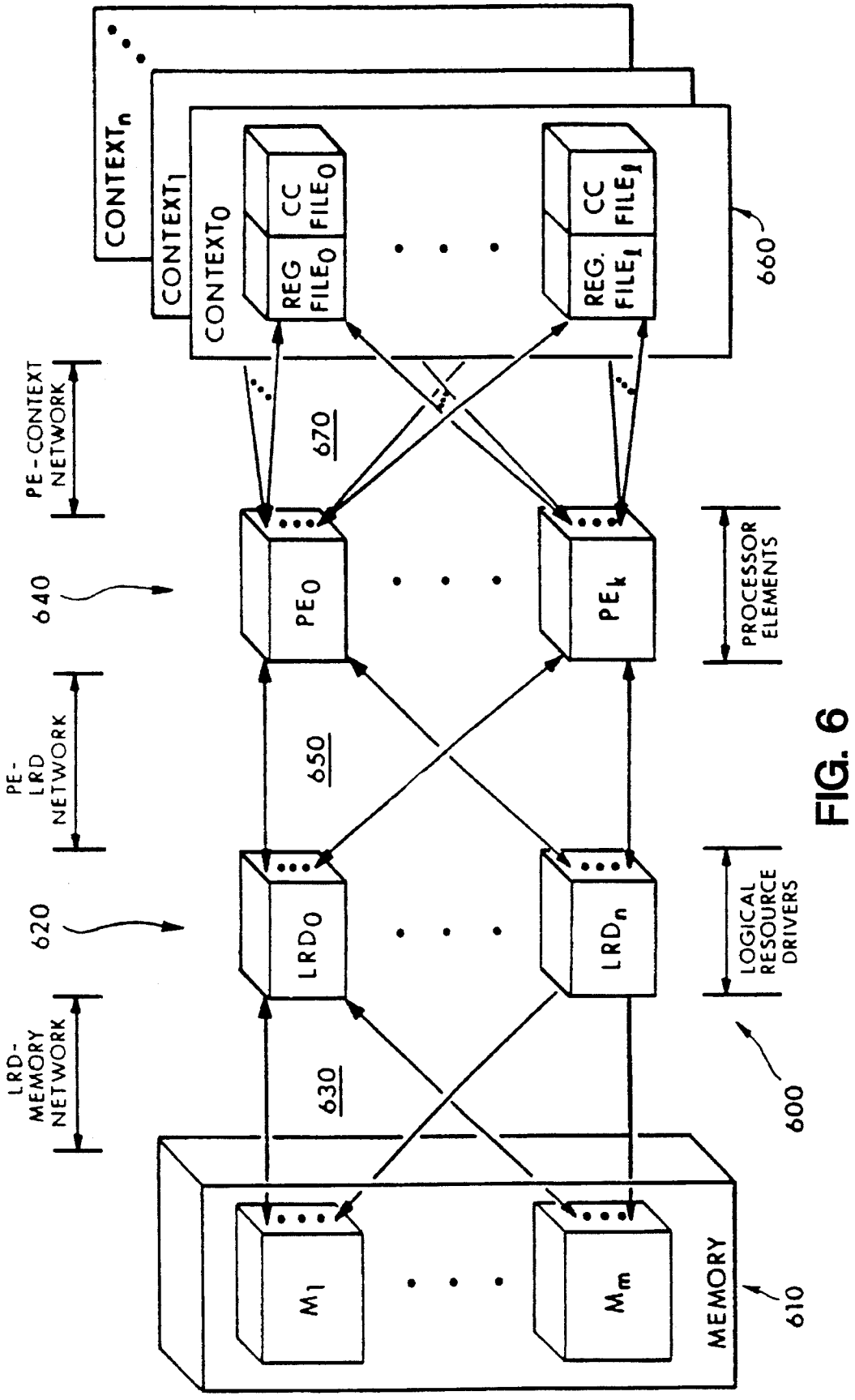

CONTEXT$_n$
CONTEXT$_1$
CONTEXT$_0$
REG FILE$_0$
CC FILE$_0$
REG. FILE$_1$
CC FILE$_1$
660
670
PE-CONTEXT NETWORK
640
PE$_0$
PE$_k$
PROCESSOR ELEMENTS
650
PE-LRD NETWORK
620
LRD$_0$
LRD$_n$
LOGICAL RESOURCE DRIVERS
630
LRD-MEMORY NETWORK
600
M$_1$
M$_m$
MEMORY
610

FIG. 6

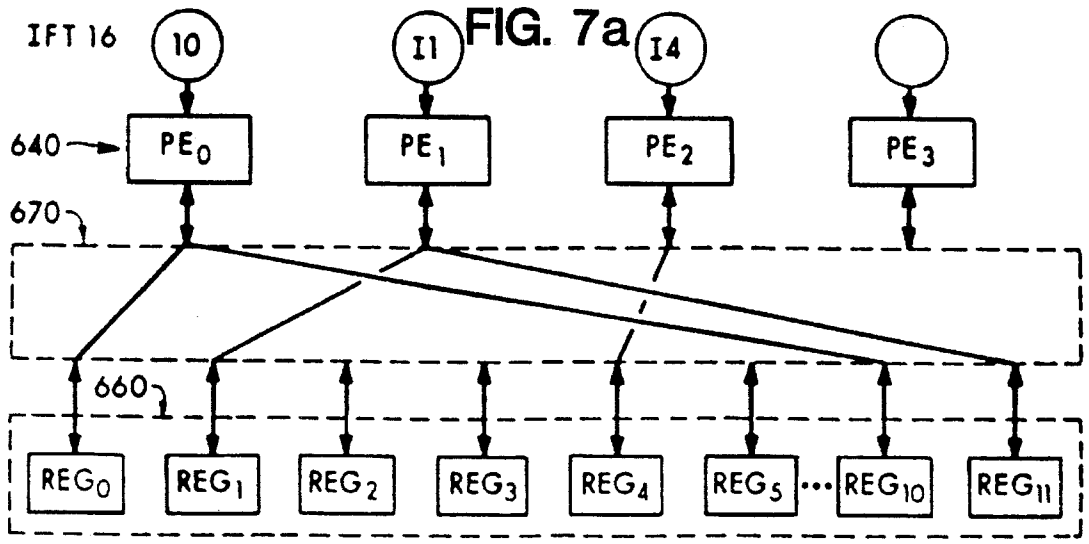


FIG. 7a

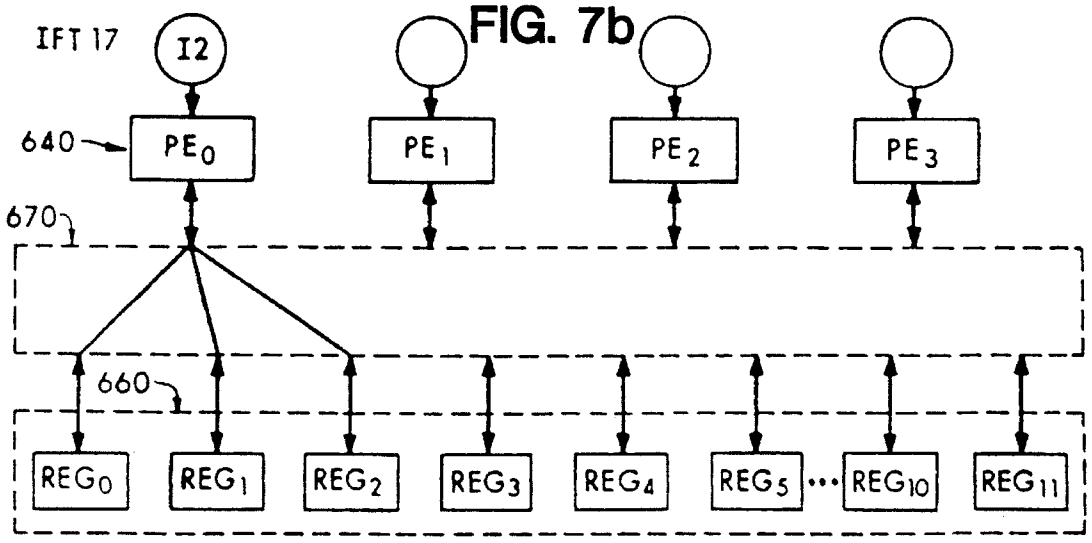


FIG. 7b

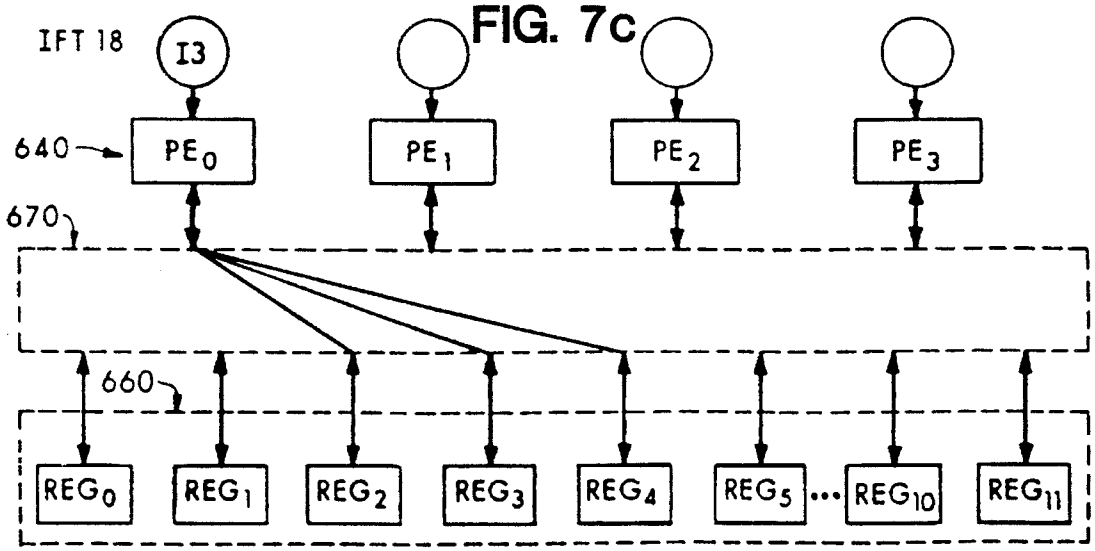


FIG. 7c

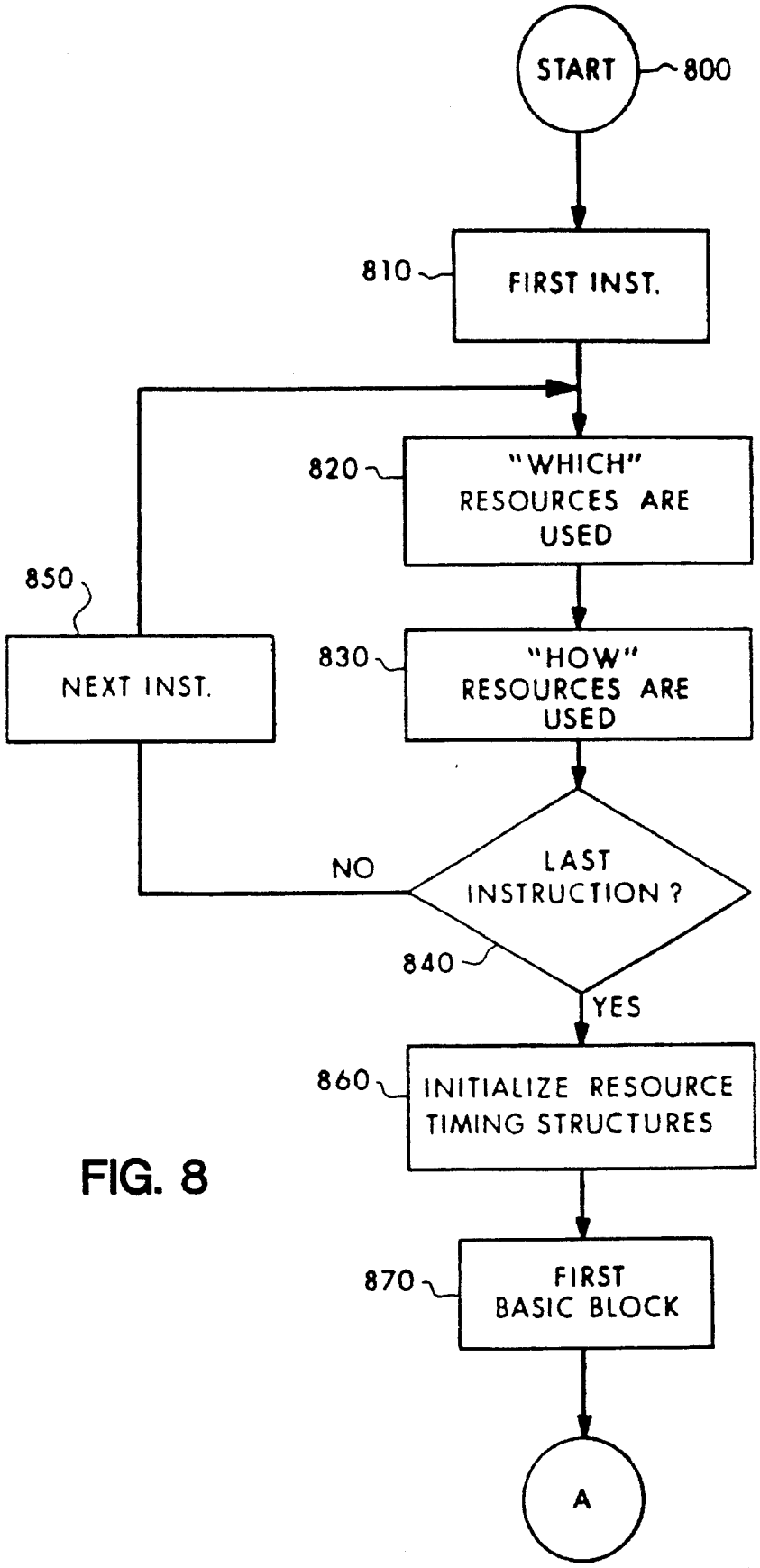

START
800
810
FIRST INST.
820
"WHICH" RESOURCES ARE USED
850
NEXT INST.
830
"HOW" RESOURCES ARE USED
NO
LAST INSTRUCTION ?
840
YES
860
INITIALIZE RESOURCE TIMING STRUCTURES
870
FIRST BASIC BLOCK
A

FIG. 8

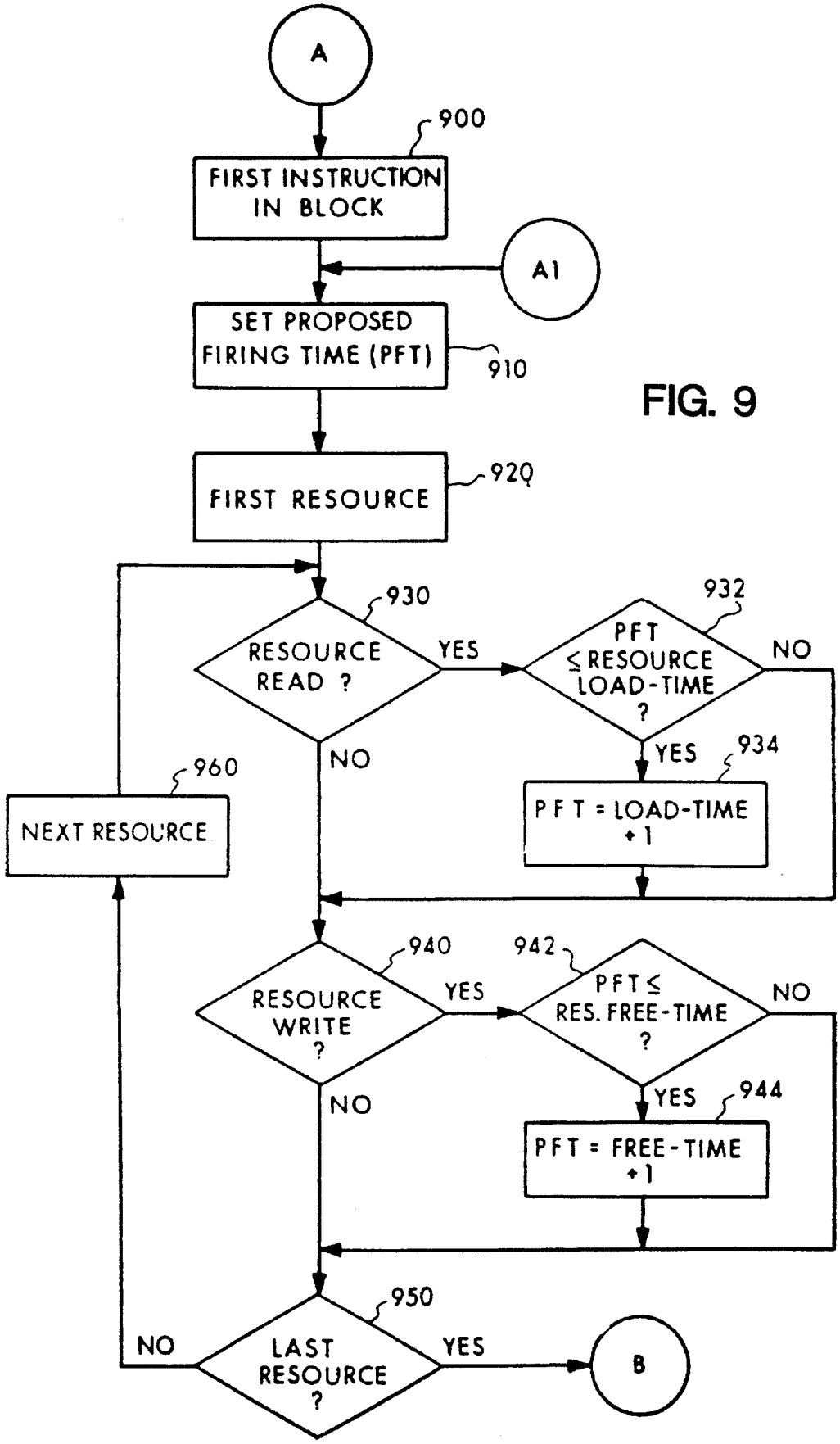

A
900
FIRST INSTRUCTION IN BLOCK
A1
SET PROPOSED FIRING TIME (PFT)
910
FIRST RESOURCE
920
930
RESOURCE READ ?
YES
932
PFT ≤ RESOURCE LOAD-TIME ?
NO
YES
934
PFT = LOAD-TIME +1
NO
960
NEXT RESOURCE
940
RESOURCE WRITE ?
YES
942
PFT ≤ RES. FREE-TIME ?
NO
YES
944
PFT = FREE-TIME +1
NO
950
LAST RESOURCE ?
NO
YES
B

FIG. 9

FIG. 10

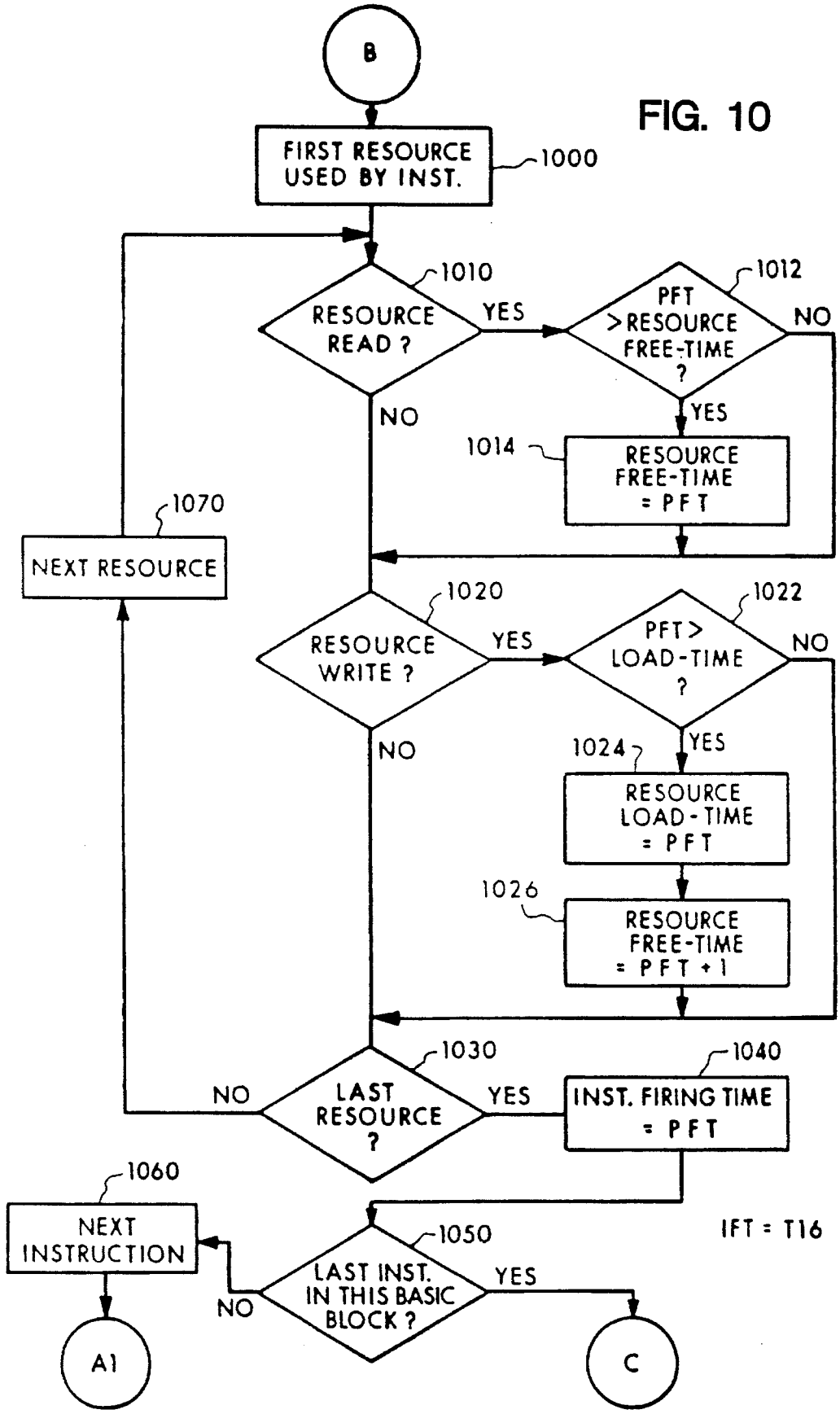
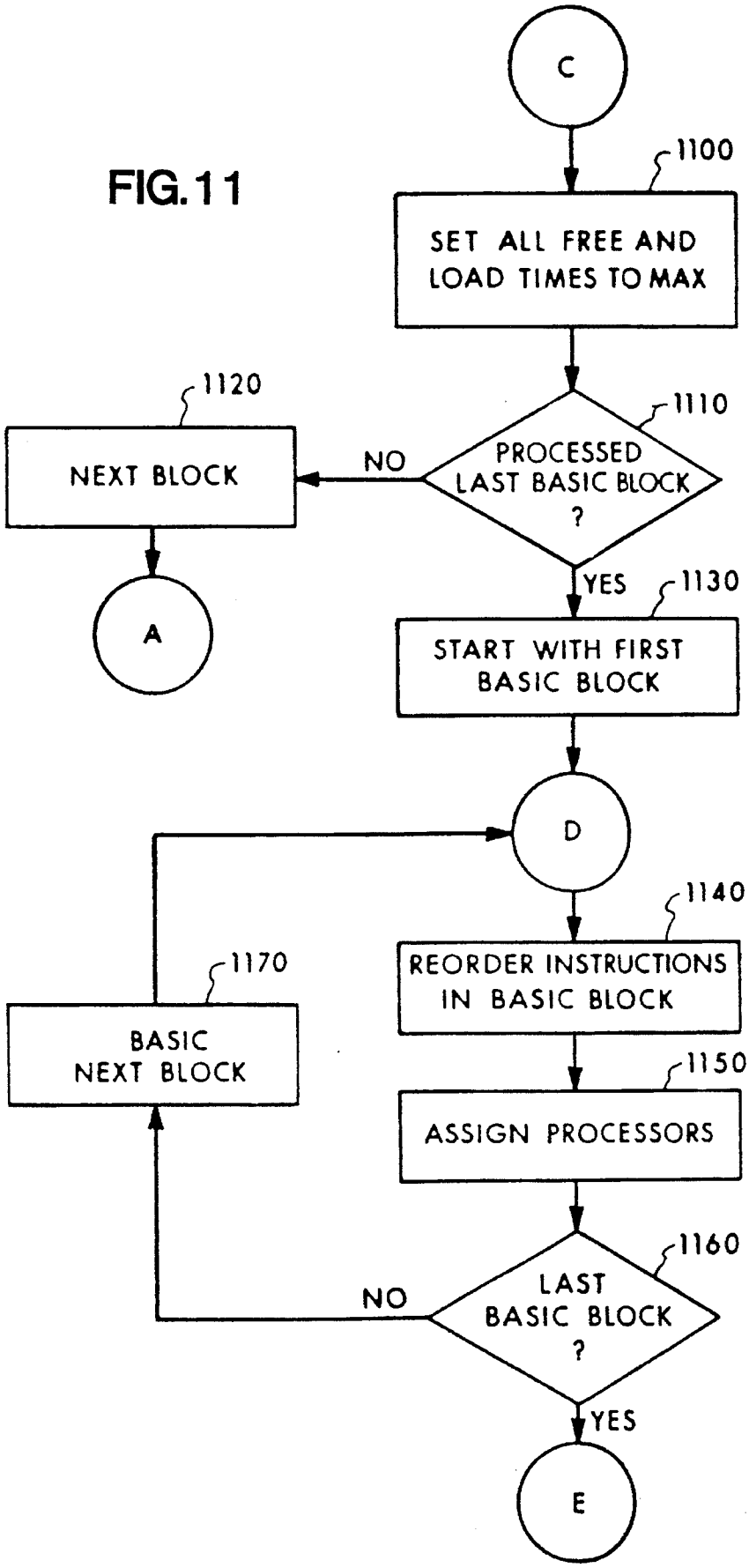

B
FIRST RESOURCE USED BY INST.
1000
1010
RESOURCE READ ?
YES
1012
PFT > RESOURCE FREE-TIME ?
NO
YES
NO
1014
RESOURCE FREE-TIME = PFT
1070
NEXT RESOURCE
1020
RESOURCE WRITE ?
YES
1022
PFT > LOAD-TIME ?
NO
NO
YES
1024
RESOURCE LOAD-TIME = PFT
1026
RESOURCE FREE-TIME = PFT + 1
1030
LAST RESOURCE ?
NO
YES
1040
INST. FIRING TIME = PFT
IFT = T16
1060
NEXT INSTRUCTION
1050
LAST INST. IN THIS BASIC BLOCK ?
NO
YES
A1
C

FIG. 11


C
1100
SET ALL FREE AND LOAD TIMES TO MAX
1110
PROCESSED LAST BASIC BLOCK ?
NO
1120
NEXT BLOCK
A
YES
1130
START WITH FIRST BASIC BLOCK
D
1140
REORDER INSTRUCTIONS IN BASIC BLOCK
1150
ASSIGN PROCESSORS
1160
LAST BASIC BLOCK ?
NO
1170
BASIC NEXT BLOCK
YES
E

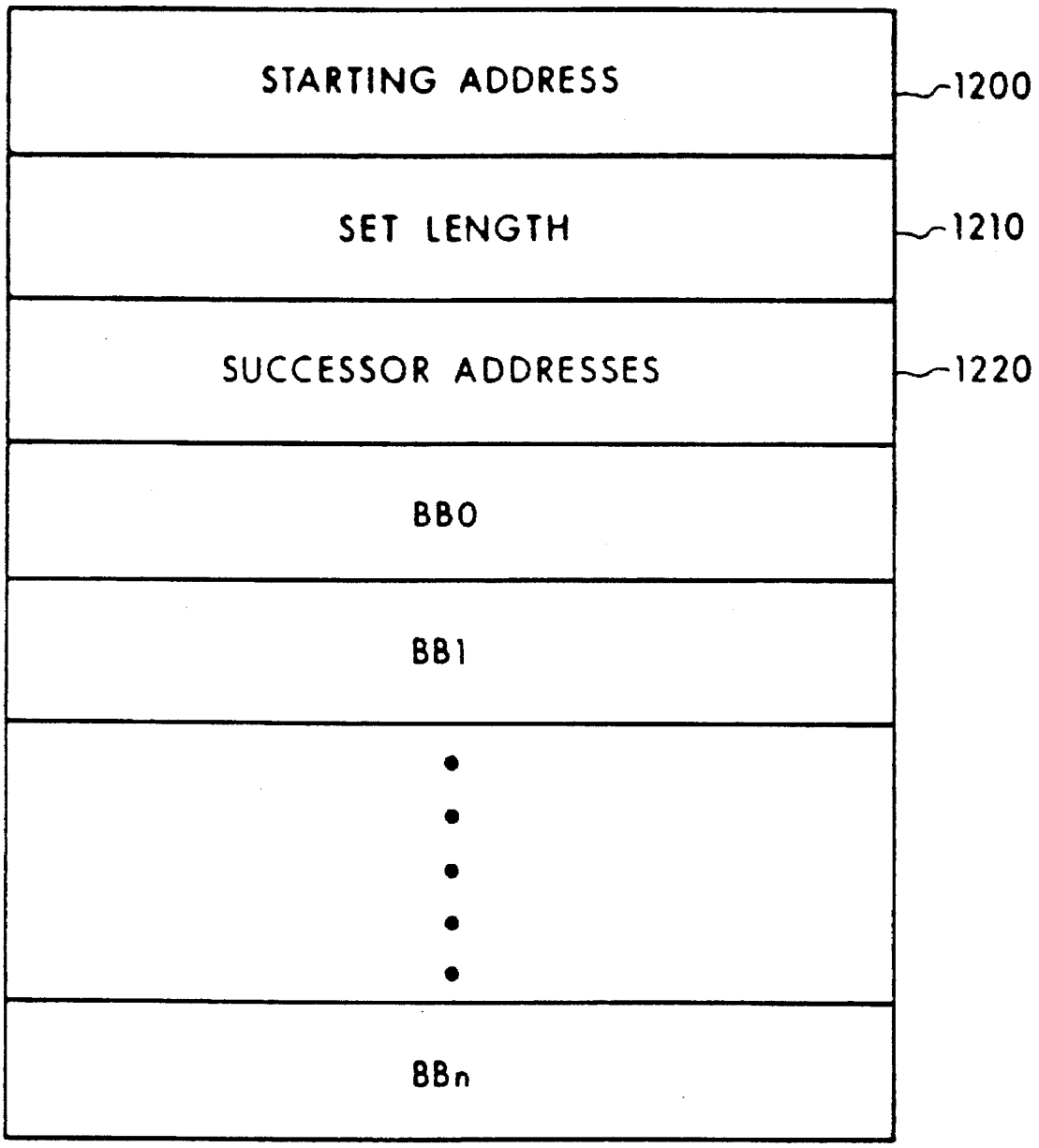

STARTING ADDRESS
1200
SET LENGTH
1210
SUCCESSOR ADDRESSES
1220
BB0
BB1
BBn

FIG. 12

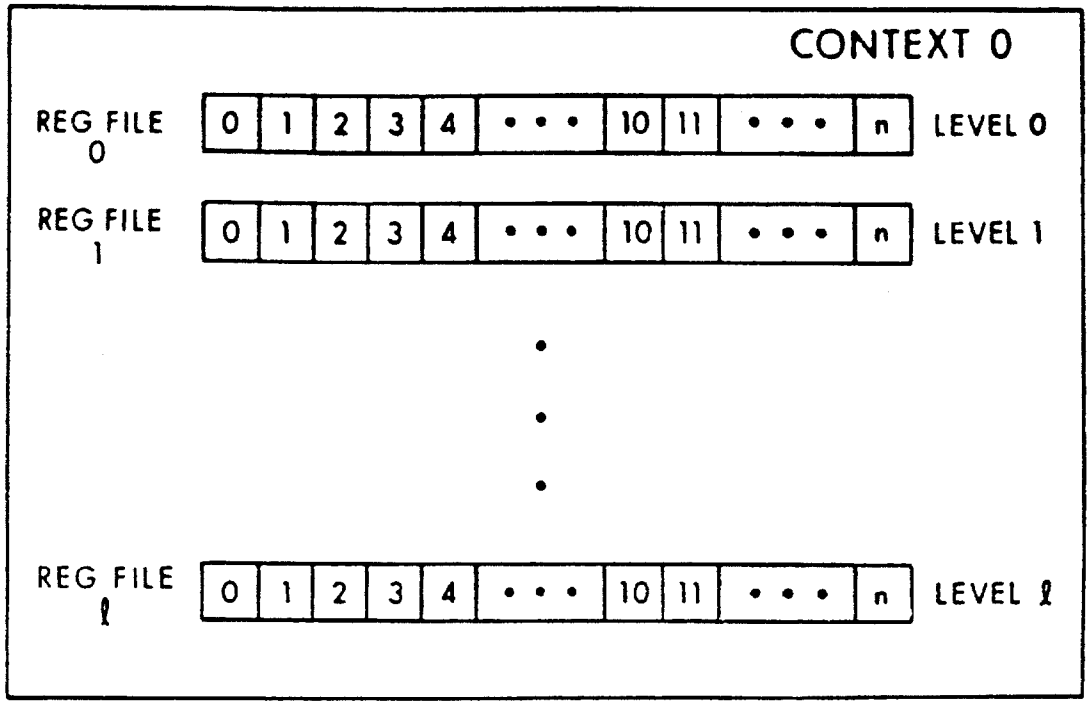

CONTEXT 0
REG FILE 0
REG FILE 1
REG FILE ℓ
0 1 2 3 4 • • • 10 11 • • • n
LEVEL 0
LEVEL 1
LEVEL ℓ

FIG. 13

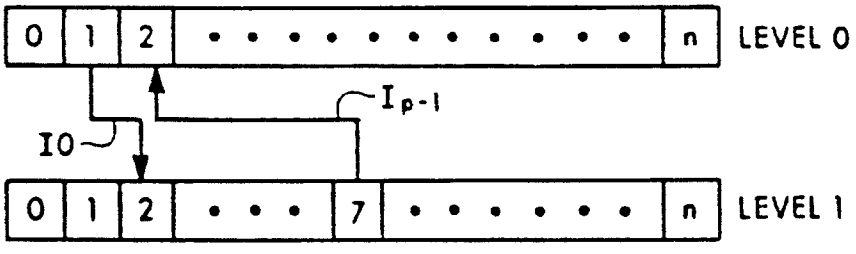

0 1 2 • • • • • • • • • • n
LEVEL 0
I_{p-1}
IO
0 1 2 • • • 7 • • • • • • n
LEVEL 1

FIG. 14

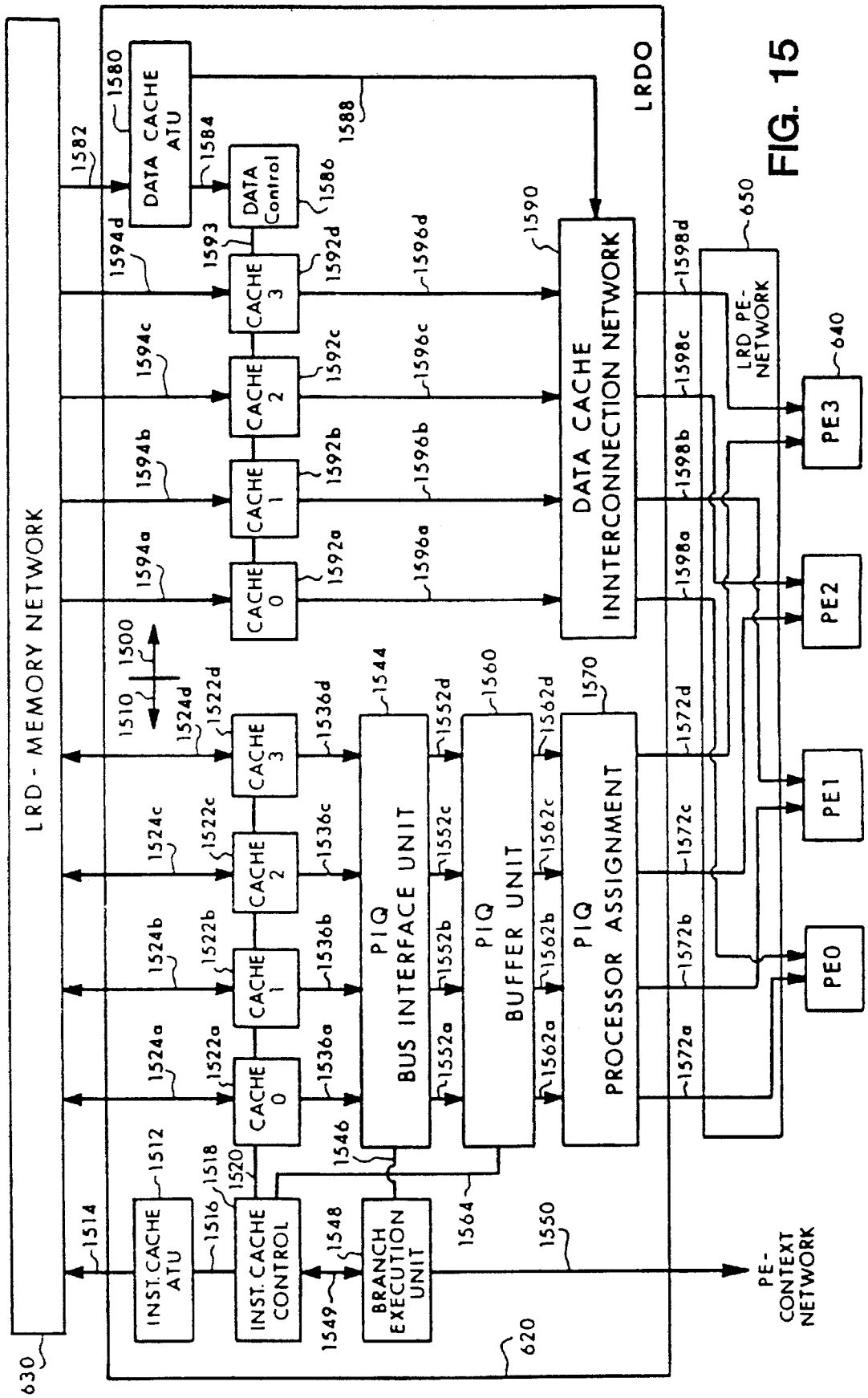

LRD - MEMORY NETWORK
630
LRDO
DATA CACHE ATU
1580
1582
1584
DATA Control
1586
1588
1590
DATA CACHE INNTERCONNECTION NETWORK
650
LRD PE-NETWORK
1598d
1598c
1598b
1598a
PE3
PE2
PE1
PE0
640
CACHE 3
CACHE 2
CACHE 1
CACHE 0
1592d
1592c
1592b
1592a
1593
1594d
1594c
1594b
1594a
1596d
1596c
1596b
1596a
1500
1510
1524d
1524c
1524b
1524a
1522d
1522c
1522b
1522a
1536d
1536c
1536b
1536a
1544
PIQ BUS INTERFACE UNIT
1552d
1552c
1552b
1552a
1560
PIQ BUFFER UNIT
1562d
1562c
1562b
1562a
1570
PIQ PROCESSOR ASSIGNMENT
1572d
1572c
1572b
1572a
1512
1514
INST. CACHE ATU
1516
1518
1520
INST. CACHE CONTROL
1546
1548
1549
BRANCH EXECUTION UNIT
1564
1550
PE-CONTEXT NETWORK
620

FIG. 15

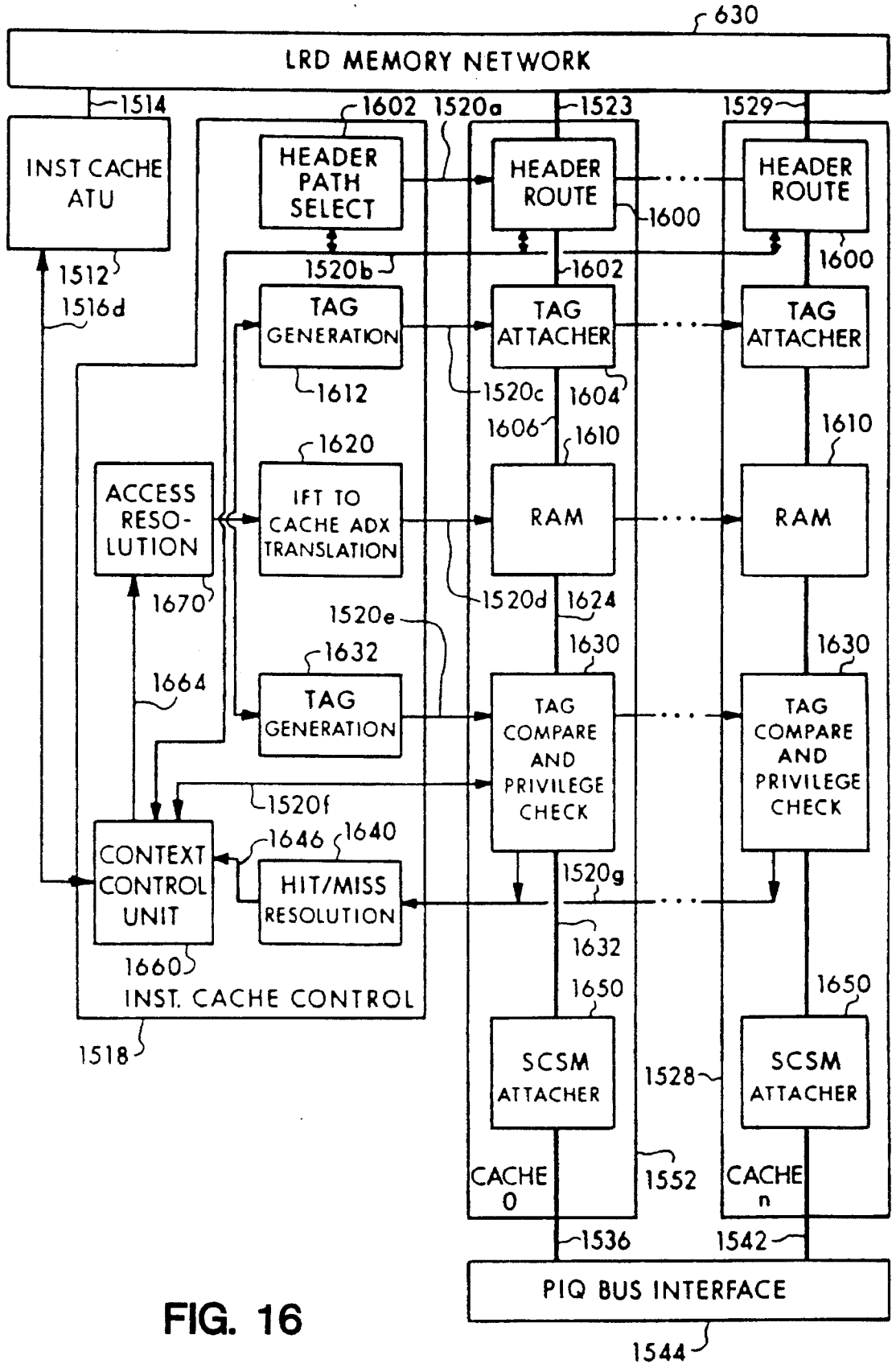

630
LRD MEMORY NETWORK
1514
INST CACHE ATU
1512
1516d
1602
HEADER PATH SELECT
1520a
1523
HEADER ROUTE
1600
1529
HEADER ROUTE
1600
1520b
1602
TAG GENERATION
1612
TAG ATTACHER
1520c
1604
1606
1610
TAG ATTACHER
1610
1620
ACCESS RESO-LUTION
1670
IFT TO CACHE ADX TRANSLATION
RAM
1520d
1624
RAM
1520e
1630
1630
1632
1664
TAG GENERATION
TAG COMPARE AND PRIVILEGE CHECK
TAG COMPARE AND PRIVILEGE CHECK
1520f
1646
1640
CONTEXT CONTROL UNIT
HIT/MISS RESOLUTION
1520g
1632
1660
INST. CACHE CONTROL
1518
1650
1650
SCSM ATTACHER
1528
SCSM ATTACHER
CACHE 0
1552
CACHE n
1536
1542
PIQ BUS INTERFACE
1544

FIG. 16

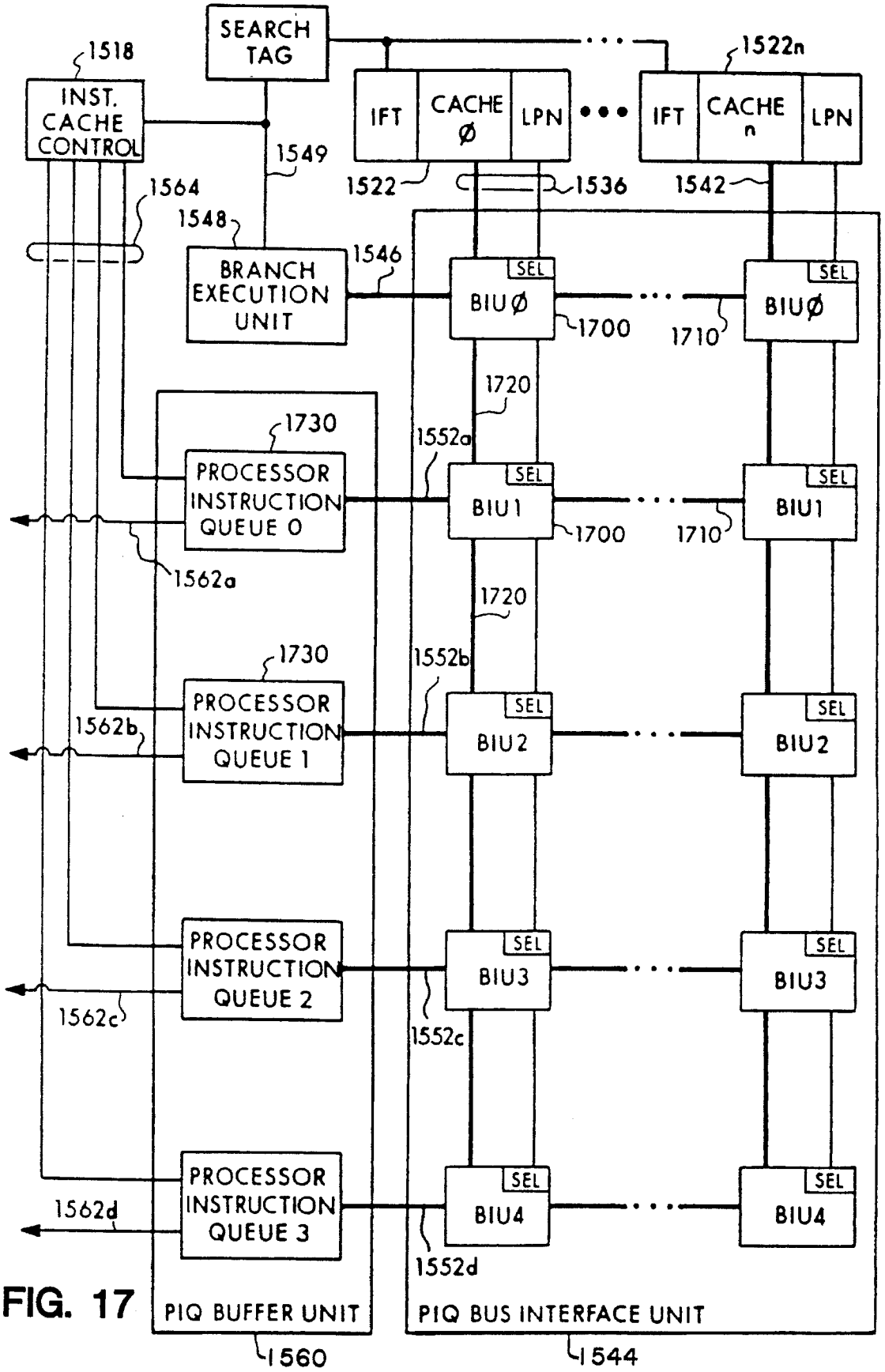

SEARCH TAG
1518
INST. CACHE CONTROL
1549
1564
1548
BRANCH EXECUTION UNIT
1546
IFT
CACHE Ø
LPN
1522
1536
1542
IFT
CACHE n
LPN
1522n
SEL
BIUØ
1700
1710
1720
1730
PROCESSOR INSTRUCTION QUEUE 0
1562a
1552a
SEL
BIU1
1700
1710
1720
1730
PROCESSOR INSTRUCTION QUEUE 1
1562b
1552b
SEL
BIU2
PROCESSOR INSTRUCTION QUEUE 2
1562c
1552c
SEL
BIU3
PROCESSOR INSTRUCTION QUEUE 3
1562d
1552d
SEL
BIU4
PIQ BUFFER UNIT
1560
PIQ BUS INTERFACE UNIT
1544

FIG. 17